Cassandra Crossing/ Attivisti della conservazione

(464)—Bibliotecari e informatici in lotta per il futuro della conoscenza e per un mondo migliore.

Cassandra Crossing/ Attivisti della conservazione

(464)—Bibliotecari e informatici in lotta per il futuro della conoscenza e per un mondo migliore.
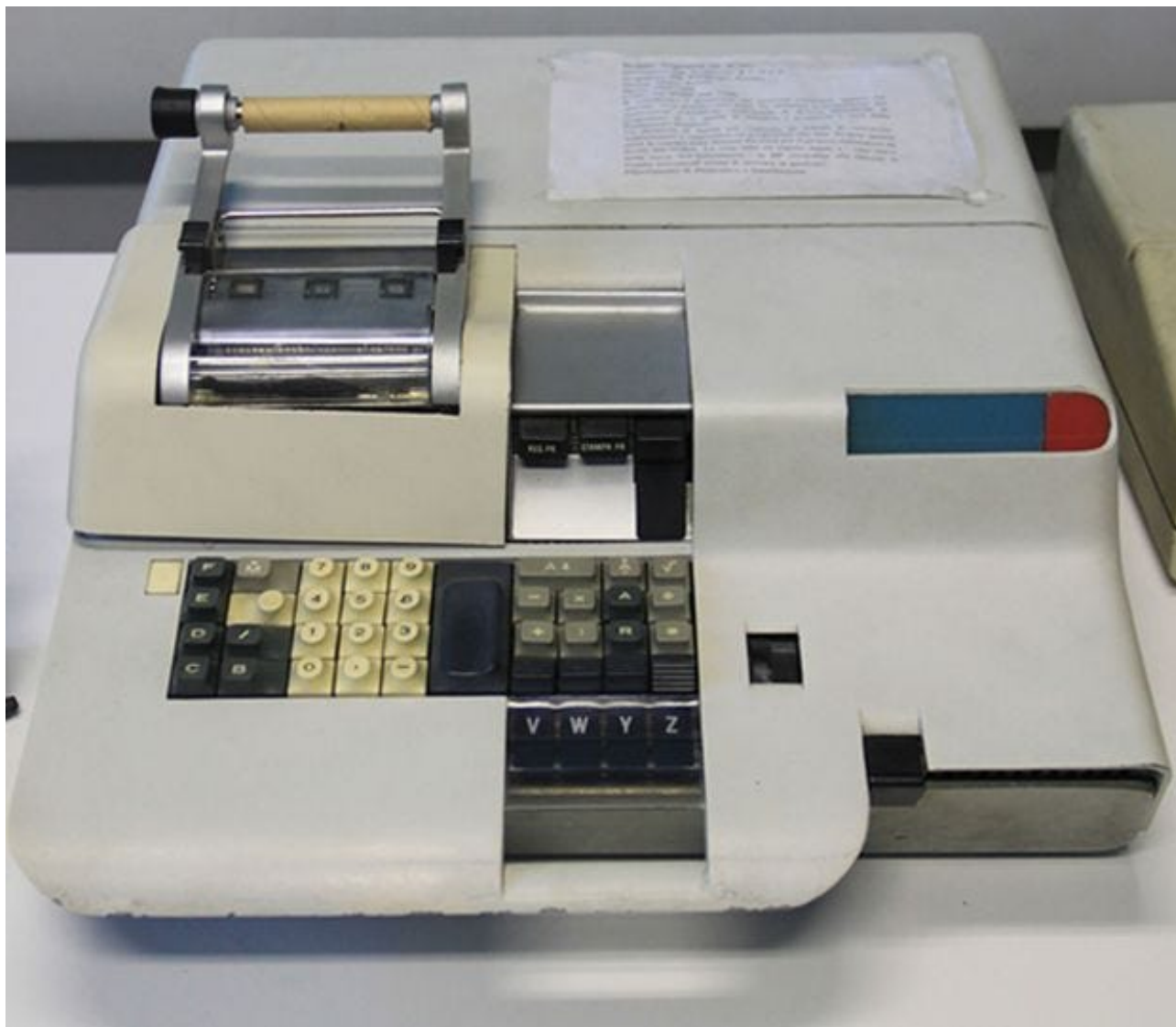
Figure 1: Olivetti Programma 101 (1965) il primo desktop computer al mondo

Il lavoro di un bibliotecario non ha certo la reputazione di essere particolarmente interessante; Cassandra, al contrario, fin da quando frequentava la biblioteca comunale ne è sempre stata affascinata. E più passano gli anni più si rende conto di quanto la conservazione dell'esistente e del passato sia una condizione necessaria per costruire il futuro.

Certo, lottare per un futuro migliore in situazioni contingenti e popolari è più facile, più interessante ed evidentemente anche più alla moda. Interessarsi, e magari addirittura preoccuparsi di cose passate, senza nemmeno la *"scusa"* di essere uno storico, sembra invece particolarmente noioso e inutile.

Al contrario, con un ragionamento che parte da lontano, proprio una di quelle digressioni che a Cassandra e ai suoi 24 irriducibili lettori piacciono tanto, si può arrivare facilmente a scoprire novità poco note ma molto interessanti, a cui potrebbe addirittura venir voglia di partecipare attivamente.

L'informatica è una scienza giovane; infatti anche se le sue radici sono vecchie di oltre due secoli, i suoi impieghi sono stati ben poco significativi dai tempi di Charles Babbage e Ada Lovelace (1822) fino alla nascita del primo computer programmabile Z3 tedesco (1941). Anche successivamente, fino a quando il computer sono rimasti comunque chiusi in pochi laboratori, la loro importanza per la scienza e l'industria è stata significativa, ma per la generalità della popolazione no.

Poi nel 1965 è nato il primo personal computer, l'italiano **Olivetti Programma 101** (P101 o); dopo la distruzione della divisione elettronica di Olivetti, fortemente voluta dalla politica italiana e americana, nel 1971 è nato il secondo "primo" personal computer (Altair 8800); ecco che due secoli abbondanti di storia dell'informatica si riducono a 49 anni.

Sì, perché la disponibilità di un computer sul tavolo di ogni studente ha permesso alla cultura di esprimersi non solo come linguaggio parlato e scritto, ma anche nella forma di una lingua nuova, la lingua parlata con il computer e tra computer per esprimere concetti e algoritmi, un *linguaggio* che da potenza diveniva atto.

Il motivo per cui la disponibilità *per tutti* di un linguaggio non solo descrittivo ma eseguibile è importante, anzi *rivoluzionario* l'ha riassunto Lawrence Lessig nella sua famosa massima *"Nel ciberspazio solo il software è legge".* Si tratta di un concetto la cui importanza è evidente, oggi che la nostra vita si svolge a metà tra mondo materiale e mondo informatico.

Sì, perché un linguaggio per computer è qualche cosa di attivo, con cui puoi trasmettere la tua volontà a un computer, che a sua volta può imporla ad altri; abbastanza per temere tante piccole Skynet, e quindi per preoccuparsi di un'etica dell'ancora in fasce intelligenza artificiale.

Che conservare lo scritto, eseguibile o meno, sia molto importante per l'umanità, è dimostrato dal fatto che da millenni questa è stata sempre considerata come una nobile, importante e necessaria attività.

Pensiamo alla biblioteca di Alessandria, distrutta dalle azioni congiunte di due religioni organizzate, e cerchiamo di capire perché diventi così importante conservare il pensiero umano prestato ai computer, scritto nel linguaggio del software.

Questo nuovo "pensiero" è più importante di altri linguaggi umani, perché diventa anche un attore, una figura attiva che può interagire da pari a pari, o qualche volta anche da superiore ad inferiore, con gli esseri umani.

Chiunque abbia utilizzato un computer, specialmente uno dei primi computer, sa perfettamente quanto il software sia purtroppo caduco, quanto corrompibile, quanto delicato, anche solo per il fatto che i computer evolvono così rapidamente; sa che i bit conservati in memorie lette e scritte da device sempre diversi e più evoluti si disperdono e si cancellano, e così il nuovo software cancella e disperde il software della generazione precedente.

Pensate ai floppy disk flessibili di un Apple][, quelli che si potevano raddoppiare di capacità facendo un buco con una punzonatrice in un punto preciso della custodia.
Pensate a tutti i supporti di memoria che li hanno sostituiti: i dischi flessibili da 320 KB, poi i dischi flessibili da 1,2 MB, poi i floppy rigidi da 1,44 MB da tre pollici e mezzo, nel loro case di plastica apparentemente tanto robusto, fino ad arrivare ai cd-rom.

Si, non dimentichiamo i cd-rom da 640 ed 800 MB, di capacità "infinita" per l'epoca, ed i cd

rom audio in cui si pensava di poter mettere tutta la musica del mondo, che sono diventati rapidamente troppo piccoli, sostituiti dai DVD da 4,3GB e dai Blu-ray da 128 GB.

Nel frattempo gli hard disk hanno cambiato completamente tecnologia almeno una mezza dozzina di volte; non sono più neanche lontanamente parenti dei primi hard disk dei mainframe; ogni volta che sono stati sostituiti, di solito quando si compra un nuovo computer, buona parte del software e delle informazioni memorizzate sui vecchi hard disk sono andati semplicemente perduti perché sottovalutati, ritenuti non importanti.

Il giornale di oggi si butta via il giorno dopo senza pensarci due volte, ma un testo miniato, frutto del lavoro di un copista del '500, non lo si butterebbe via con la stessa facilità; oggi in effetti il software memorizzato sul supporto di una vecchia periferica ricade, nella vita quotidiana, proprio nella categoria dei giornali, delle cose vecchie, anzi invecchiate, di cui si può fare da meno, e questo è un processo continuo per cui il software continua a essere scritto e gettato via.

E questo vale ovviamente anche per tutto quello che è digitale, quindi anche immagini, musica testi, registrazioni di audio e video, intere opere multimediali legate ai sistemi operativi defunti, e quindi ora illeggibili e perdute per sempre.

Ecco che a questa perversione del software che viene perso perché sottovalutato, se ne affianca un'altra, causata dalla continua e frenetica evoluzione delle periferiche di memorizzazione, un altro effetto perverso per cui il linguaggio dei computer, il meglio della produzione della mente delle persone, si perde.

Il Bit sembra una cosa eterna, ma se si va a vedere quello che è successo e succede nella realtà è quanto di più caduco e deperibile ci sia.

Pensiamo alle tavolette di argilla che sono giunte fino a noi attraverso i millenni, pensiamo alla carta pergamena dello scorso millennio, od anche solo alla buona e durevole carta del secolo scorso, prodotta magari in maniera molto inquinante, ma che è arrivata fino ad oggi.
Tutte questi supporti per la scrittura si sono rivelati molto più durevoli, decenni, secoli o millenni, delle più avanzate tecnologie informatiche, che hanno causato la perdita di enormi quantità di dati e di software ad ogni cambio di generazione.

Per fortuna esistono anche degli informatici che di conservazione dei dati si devono preoccupare, principalmente perché sono responsabili di grosse basi di dati di grande importanza ed enorme valore economico.
Certo, quello che fanno Google, Amazon e gli altri Cloud su questo problema non è noto ai più; lo è invece, per esempio, quello che fa la NASA, che essendo un ente pubblico rende (spesso) pubblico il suo operare.

Si è scoperto così che uno dei metodi più efficaci per conservare dati e software è lasciarli sui vecchi nastri magnetici di una volta o comunque su nastri magnetici un po' più evoluti e più capaci come le cassette digitali.

Conservare ogni nastro in almeno in due copie dentro "librerie digitali", che sono in grado di immagazzinare e gestire migliaia di nastri, ed automaticamente, sotto il controllo di un software apposito, copiare periodicamente i dati da un nastro all'altro in modo che la frequenza di copiatura sia maggiore del tempo in cui un nastro diventa illeggibile.
Infatti i nastri magnetici ,anche con vita prevista intorno alla trentina d'anni, in realtà già dopo 10 anni possano diventare illeggibili

Esistono poi iniziative molto più ambiziose, che partono dall'idea di preservare il software per l'eternità o quasi, come il progetto **GitHub Archive Program** per la conservazione di una copia degli immensi archivi di software di Github.com nel gelo delle isole Svalbard.

In questo progetto, oltre a cercare il freddo ed un supporto dati stabile e durevole, ci si è dovuto porre il problema del futuro recupero dei dati da parte di persone che non avranno minimamente accesso ai computer ed alle periferiche attuali.

Questo ha portato a "inventare" un modo vecchio e banale di registrare i dati su un supporto; è un nastro di materia plastica particolare, la cui durata si stima superiore ai 100 anni, che contiene fori e non-fori.
Ha richiesto anche di concepire e realizzare dei metadati e delle meta-informazioni, in modo da permettere, a chi dovesse ritrovare questi nastri, di costruire un apparecchiatura per rileggerli nella maniera più rapida ed efficiente, e poter recuperare così i dati originali

Questo è proprio ciò che generazioni e generazioni di amanuensi e bibliotecari hanno fatto da millenni, mantenendo gli indici dei testi, conservandoli il meglio possibile, spolverandoli, catalogandoli, copiandoli, sostituendoli quando venivano danneggiati e così via.

Esistono altri progetti moderni che si pongono il problema della conservazione del software e dei materiali digitali per i posteri.

Il primo è l'arcinoto **Internet Archive** (The Wayback Machine); si occupa "in primis" di prendere periodiche "fotografie" di siti web noti su internet e di archiviarle permanentemente, in modo da mantenere la memoria sia di un sito che della sua evoluzione nel tempo.

E' una risorsa importante perché, anche se non archivia in modo perfetto i siti (non può per esempio salvare bene i siti attivi per via degli script che contengono) può comunque salvare i contenuti testuali e grafici, fino al punto da servire come risorsa autorevole.
Nei tribunali e nelle questioni legali ad esempio, viene talvolta utilizzata per stabilire chi ha ragione in procedimenti che abbiano a che fare con la proprietà intellettuale o più semplicemente col comportamento della gente in rete.

In Internet Archive esistono anche altri progetti estremamente interessanti, che si pongono, in maniera diversa, lo stesso obiettivo di conservazione della conoscenza.

Internet Archive infatti realizza un archiviazione massiccia di libri e di altri tipi di supporti, di software su floppy, di videogiochi, e digitalizza i supporti analogici in modo da poterli conservare in forma digitale, archiviandoli in maniera ridondante non nel cloud ma su dei server dedicati.
A proposito, i server non sono gratuiti, mandategli qualche soldo !

Ricordiamoci delle risorse che tutti i giorni utilizziamo su internet, e che non sono fornite da Google o di Facebook, ma da persone che le creano e le fanno funzionare per passione o per missione; Wikipedia, Internet Archive, il Progetto Gutenberg (ancora assurdamente censurato in Italia), il Progetto Manunzio; **questa gente lavora per voi e per il vostro futuro, lo fa per passione ed ha bisogno di soldi; anche pochi euro di donazione per loro sono vitali. Capitelo.**

Ma torniamo al **GitHub Archive Program** ed al problema di "rimediare" all'evoluzione e all'obsolescenza dei computer, utilizzando un supporto di memorizzazione dati creato apposta.

Come già detto, si tratta di un nastro di una materia plastica particolarmente durevole, almeno 100 anni, su cui la registrazione viene effettuata in maniera simile a quella delle antiche schede o nastri perforati; per la decodifica di questo nastro non esiste una periferica commerciale; ma insieme agli archivi di software vengono memorizzate anche le informazioni necessarie perché il futuro scopritore di questo archivio possa costruirne una equivalente con le tecnologie del suo tempo.

Tuttavia il software memorizzato, per essere di qualche utilità, dovrebbe poter funzionare su un computer, computer che ormai non esiste più.

Si tratta quindi di rendere il software eseguibile anche per gli "informatici del futuro", ed il software, specialmente quello scritto non al giorno d'oggi ma qualche anno fa, non è facilmente trasportabile da un computer ad un altro.
Questo perché il software una volta si affidava molto alle peculiari caratteristiche del computer e del sistema operativo su cui girava, per cui non può essere usato se di questo hardware e sistema operativo non si dispone.

Sembra un problema senza soluzione, mentre invece una soluzione esiste e si chiama "emulazione".
Molti di noi hanno usato gli emulatori di vecchi computer, come l'emulatore dell'**Apple][europlus** oppure il progetto **MAME**, che emula una antica e diffusissima piattaforma di videogame da sala.
Questi emulatori sono esempi di un singolo programma che risolve il problema di eseguire tutto il software sviluppato per un certo hardware, ormai non più disponibile.

Ovviamente l'emulatore di un computer è a sua volta un software, e si ricrea quindi la necessità di farlo funzionare su un computer del futuro; tuttavia la dimensione del problema si riduce grandemente, perché un solo emulatore può eseguire tutto il software scritto per un computer, e perché i sorgenti dell'emulatore potrebbero essere scritti con particolare cura in modo da renderlo più facilmente modificabile, interpretabile e installabile su un futuro computer.
Il software di emulazione potrebbe essere archiviato "in evidenza" rispetto al resto, in modo da far entrare anche il vecchio software in formato eseguibile in questo gioco virtuoso del software conservato per i posteri. Un idea per un nuovo progetto di conservazione? Od una copia software di "Contact"?

Ma il problema della conservazione del software "vecchio" non si limita all'esecuzione di programmi in formato eseguibile, anche perché la maggior parte del software sviluppato al giorno d'oggi realizza solo un'unica funzionalità, che è molto meno della funzionalità di un'intera applicazione.

Infatti di solito il software viene sviluppato in forma di "librerie"; la maggior parte dei software che è stato scritto negli ultimi due decenni è in forma di librerie di subroutine, e non di programmi applicativi. Gli applicativi vengono costruiti assemblando le librerie adatte, scrivendo solo una quantità di codice molto limitata.

Quando uno sviluppatore produce il software di una libreria, deve obbligatoriamente porsi il problema di renderlo; interpretabile da chi lo utilizzerà; per fare questo deve realizzare, e sopratutto documentare, una API "*application programming interface*", cioè una interfaccia applicativa da utilizzare per fornire alla libreria i dati di input in modo ben documentato, ottenendo i risultati sempre in un formato standard e ben descritto.

Questo è di per sé un modo molto efficace per favorire il riutilizzo del software perché, una volta che si sia risolto il problema di disporre dell'hardware o di un suo emulatore, l'utilizzo di software che preveda l'impiego di una API diventa particolarmente semplice, tanto è vero che l'informatica moderna ha "naturalmente" adottato questo modo di lavorare.

A questo punto, oltre a trovare il modo di conservare i bit per un lontano futuro, bisogna risolvere i due problemi che si presentano quando i bit che conserviamo sono software.

Il primo è quello di far girare una vecchia applicazione in formato eseguibile su un emulatore; questo risolve il problema del riutilizzo dell'applicazione completa, cosa non futuribile ma che si faceva anche nel passato.

Ricordo perfettamente quando sui nuovissimi IBM AS400 venivano installati i programmi applicativi; l'AS400 in realtà eseguiva un emulatore del suo predecessore, il System 38, dentro il

quale girava un emulatore del suo predecessore, il System 36, dentro il quale girava l'antico (e perfettamente funzionante) programma di contabilità scritto e compilato per il System 36.

Il secondo problema, cioè il riutilizzo del software in formato sorgente per costruire nuove applicazioni, oggi come in un remoto futuro, vuol dire riuscire ad riutilizzare un software (una libreria), che risolve un singolo problema, come parte per scrivere una nuova applicazione.

Il progetto **Software Heritage**, che è una collaborazione accademica internazionale, nata in Francia e con contributi anche italiani, non si interessa direttamente alla conservazione a lungo termine dei codici sorgenti che descrivono i programmi ad alto livello, ma piuttosto alla "trasformazione" dei codici sorgenti stessi, scritti in forma poco riutilizzabile, in modo che siano più facilmente comprensibili e riutilizzabili.

Questo viene realizzato in maniera volontaria, mettendo a disposizione di quegli sviluppatori che intendono preservare e rendere riutilizzabile il loro software, od un software open source e libero realizzato da altri, la possibilità di farlo eseguendo una serie di passi e di procedure standardizzate messe a punto dai ricercatori del progetto Software Heritage.

In questo modo gli sviluppatori possono trasformare un software "normale", destinato solo a realizzare un'applicazione, in un software facilmente riutilizzabile perché scritto in maniera standardizzata, dotato di API e di altre particolarità, in modo che possa essere riutilizzato con la massima semplicità come libreria di nuove applicazioni.

Apprezzare pienamente lo scopo del progetto Software Heritage non è così semplice come comprendere quello di The Wayback Machine o di GitHub Archive Program; forse può essere apprezzato in pieno solo da chi abbia passato una vita professionale scrivendo, od anche semplicemente manutenendo, grossi software in ambiente professionale.

Riassumendo. Di tutti questi passi per la conservazione dei bit, e di altri ancora da venire, c'è un grande, ma poco percepito, bisogno.

Sì, perché se mai preservare il software per il domani od addirittura per le generazioni future diverrà importante, questi progetti ci condurranno verso l'obbiettivo di un software che sia praticamente eterno, come la sua natura digitale gli permetterebbe, per essere di maggiore aiuto all'umanità.

Cassandra è personalmente orgogliosa che esistano persone ed organizzazioni che, oltre a scrivere software libero ed open source, si preoccupino anche di questi problemi, poco attraenti e popolari, e che invece sono il futuro della conoscenza proprio come le biblioteche lo sono state in passato.

Non siamo eterni, e tra anni o decenni soltanto pochi bit in rete ricorderanno chi e che cosa eravamo.
Saranno bit contenenti testi, immagini, video ma anche programmi.
E ripetiamolo, i programmi non sono solo informazioni, sono la legge e la struttura del ciberspazio, che solo di software è costituito.

E' davvero bello pensare che si stia costruendo una strada che, forse, riuscirà a traghettare verso il futuro la parte più importante del "noi" digitale.

Originally published at https://www.zeusnews.it.

By Marco A. L. Calamari on November 9, 2020.

Canonical link